

Requested Patent: JP11045228A

Title:

APPLET REDIRECTION FOR CONTROLLED ACCESS TO NON-ORIGINATING  
HOSTS ;

Abstracted Patent: US5987523 ;

Publication Date: 1999-11-16 ;

Inventor(s):

HIND JOHN RAITHEL (US); LINDQUIST DAVID BRUCE (US); NANAVATI PRATIK  
BIHARILAL (US); TAN YIH-SHIN (US); WESLEY AJAMU AKINWUNMI (US) ;

Applicant(s): IBM (US) ;

Application Number: US19970868611 19970604 ;

Priority Number(s): US19970868611 19970604 ;

IPC Classification: G06F9/06 ; G06F13/14 ;

Equivalents: CN1210308

ABSTRACT:

A method and apparatus for allowing dynamic applet access to servers from which the applet did not originate wherein an application on the originating server redirects communications between the applet and network resources.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-45228

(43) 公開日 平成11年(1999) 2月16日

(51) Int. Cl. <sup>6</sup>	識別記号	F I
G 0 6 F 15/00	3 1 0	G 0 6 F 15/00 3 1 0 D
9/44	5 3 0	9/44 5 3 0 M
13/00	3 5 5	13/00 3 5 5

審査請求 有 請求項の数 2 O L (全 8 頁)

(21) 出願番号 特願平10-145371

(22) 出願日 平成10年(1998) 5月27日

(31) 優先権主張番号 08/868611

(32) 優先日 1997年6月4日

(33) 優先権主張国 米国 (U S)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州

アーモンク (番地なし)

(72) 発明者 ジョン・レイテル・ハインド

アメリカ合衆国27613 ノースカロライナ州ローリー ハリントン・グローブ・ドライブ 5408

(74) 代理人 弁理士 坂口 博 (外1名)

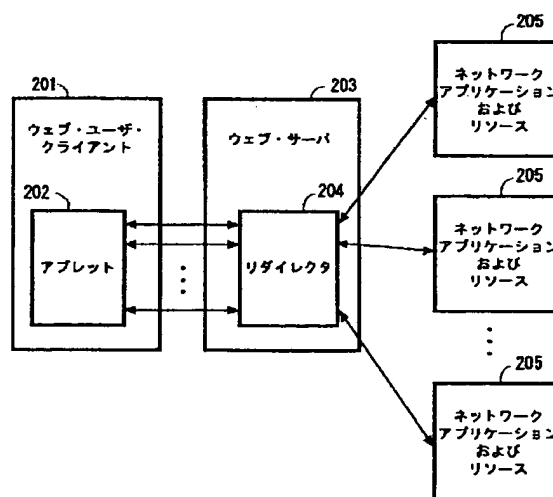
最終頁に続く

(54) 【発明の名称】 オブジェクトのリダイレクト管理方法及びアプレットの通信を可能とする方法

(57) 【要約】

【課題】 アプレットが発信されたものではないサーバへの動的アプレット・アクセスを可能とし、発信元サーバ上のアプリケーションがアプレットとネットワーク・リソースの間の通信をリダイレクトする方法及び装置。

【解決手段】 各々が一意のアドレスによって識別されるユーザ・ワークステーション及びホストのネットワークにおけるプログラムされたオブジェクトのリダイレクトを管理する方法において、アクセス対象のホストのアドレス及び前記ホストの各々にアクセスすることを認められているユーザのアドレス範囲を含んでいるホスト・アクセス・フィルタリング・テーブルを作成するステップと、前記ユーザが前記フィルタリング・テーブルにプログラム式に前記ユーザからのアクセスが認められているホストへアクセスするステップとを備えている方法。



**【特許請求の範囲】**

【請求項1】 各々が一意のアドレスによって識別されるユーザ・ワークステーション及びホストのネットワークにおけるプログラムされたオブジェクトのリダイレクトを管理する方法において、

アクセス対象のホストのアドレス及び前記ホストの各々にアクセスすることを認められているユーザのアドレス範囲を含んでいるホスト・アクセス・フィルタリング・テーブルを作成するステップと、

前記ユーザが前記フィルタリング・テーブルにプログラム式に前記ユーザからのアクセスが認められているホストへアクセスするステップとを備えている方法。

【請求項2】 アプレットをダウンロードしたホスト以外のホストとの前記アプレットの通信を可能とする方法において、

アプレットを一意のアドレスを有しているウェブ・サーバからユーザ・ワークステーションにダウンロードするステップと、

前記ウェブ・サーバのアドレスを検出するステップと、

前記ウェブ・サーバからリダイレクタ・ポート番号へのソケット接続を開くステップと、

前記リダイレクタから前記ユーザ・ワークステーションへターゲット・ホストに対するアドレスを送るステップと、

前記ターゲット・ホストと前記アプレットを実行する前記ユーザ・ワークステーションとの間のソケット接続を開くステップとを備えている方法。

**【発明の詳細な説明】****【0001】**

【発明の属する技術分野】 本発明はコンピュータ・システムに関し、ネットワークに接続されたこのようなコンピュータ・システムを操作して、コンピュータ・システムがネットワークによってデータ及びコードにアクセスできるようにする方法に関する。具体的にいえば、アプレットにより、これらの発信元ではないホストとの通信に関する。

【0002】 関連出願1997年6月4日出願で、インターナショナル・ビジネス・マシーンス・コーポレーションに譲渡された「Host Information Access via Distributed Programmed Objects」なる名称の米国特許第868873号。

**【0003】**

【従来の技術】 過去数年間で、インターネット、詳細に言えばインターネットの上に設けられた機構の1つであるワールドワイド・ウェブ(WWWないしWeb)が爆発的に成長してきている。WWWは多くの異なるサーバに分散された多くのページないしファイルの情報を含んでいる。各ページはユニバーサル・リソース・ロケータ(URL)によって識別される。URLはサーバ・マシン

とそのマシン上の特定のファイルないしページの両方を指定する。単一のサーバには多数のページないしURLが常駐していることができる。

【0004】 WWWを使用するために、クライアントはWeb Explorer (オペレーティング・システム/2 (OS/2) (c) IBMコーポレーション) または Netscape Communications Corporation から入手可能な Navigator (c) プログラムなどのWebブラウザといわれるソフトウェアを実行する。クライアントはブラウザと対話をして、特定のURLを選択し、これによりブラウザはそのURLないしページに対する要求をURLで特定されたサーバに送る。通常、サーバは要求されたページを検索し、そのページに関するデータを要求元のクライアントへ送り返すことによって、要求にこたえる(クライアント・サーバの対話はハイパーテキスト・トランスポート・プロトコル(「HTTP」)にしたがって行われる)。このページが次いでクライアントの画面でユーザに対して表示される。クライアントはサーバにアプリケーションを起動させて、たとえば、特定の話題に関するWWWページを探索することも行う。場合によっては、ユーザに対するアクセスを選別して、特権ユーザだけに情報にアクセスすることを認める、ファイアウォールなどのセキュリティ機構のため、サーバに連絡を取れないことがある。これらの場合には、プロキシ・サーバまたはプロキシ・アプリケーションを使用して、このようなアクセスを管理させることもできる。プロキシ・サーバは保護及び非保護ネットワーク域にまたがり、関与しているユーザ及びこれらのユーザに対して構成されている特権に基づいてこれらの域の間の通過トラフィックを容易とするエンティティとみなすことができる。使用されるネットワーク接続はソケットと呼ばれ、これはネットワークからのデータストリームがどこで送受信されるかに過ぎない。サーバの番号付のポートを開いて、特定のソケットのデータストリームを聴取することができる。

【0005】 ほとんどのWWWページはHTML (ハイパーテキスト・マークアップ言語) という言語にしたがってフォーマットされる。それ故、典型的なページはテキストを、タグと呼ばれる埋め込みフォーマット・コマンドとともに含んでおり、タグを使用して、フォント・サイズ、フォント・スタイル (たとえば、イタリックか太字か)、テキストをどのようにレイアウトするか、及びその他のページのオプションを制御することができる。WebブラウザはHTMLスクリプトを分析して、指定されたフォーマットにしたがってテキストを表示する。さらに、HTMLページは、マルチメディア・データ、たとえば画像、ビデオ・セグメントまたはオーディオ・ファイルに対する、他のURLによる参照も含んでいる。Webブラウザはデータを検索し、これを表示す

たは再生することによってこのような参照に応答する。あるいは、このようなマルチメディア・データは、周囲のHTMLテキストなしで、それ自体のWWWページを形成することもできる。

【0006】ほとんどのWWWページは他のWWWページに対する1つまたは複数の参照も含んでいるが、これらはもとのページと同じサーバにある必要はない。このような参照は一般に、画面上の特定の位置を選択するユーザにより、通常は、マウス制御ボタンを(ダブル)クリックすることによって活動化される。これらの参照またはロケーションはハイパーリンクとして知られており、通常、特定の態様でブラウザによってフラグがつけられている(たとえば、ハイパーリンクに関連付けられたテキストは他の色であってもよい)。ユーザがハイパーリンクを選択した場合、参照付きのページが検索され、現在表示されているページと置き換わる。

【0007】HTML及びWWWに関する詳細な情報はDouglas McArthurの「World Wide Web and HTML」、18~26ページ、Dr Dobbs Journal、1994年12月及びIan Grahamの「The HTML SourceBook」(John Wiley, New York, 1995年)に記載されている。

【0008】以上で述べ、また広い意味で、現在実施されているWWWには、サーバからクライアントにダウンロードされたページが本質的に受動的なものである、換言すると、これらがクライアント・マシンで実行されるコードを含んでいないという欠点がある。これが意味することの1つは、サーバがクライアントとサーバの間の対話に関連する処理をクライアントへオフロードできないということである。それ故、クライアントが、たとえば、電話番号の入っている書式を作成している場合、電話番号の桁数に関する正式なチェックはサーバで行わなければならない。これはまずサーバに高い処理負荷をもたらす、次いで、何らかの修正すべき誤りがあった場合に、サーバとクライアントの間での時間のかかる余分な通信をもたらすものである。さらに、サーバがクライアントでの実行のためにコードをダウンロードできないことは、WWWを活用するために作成できるアプリケーションのタイプに対して大きい制限となる。

【0009】特にSun Microsystems Inc. のJava (c)Sun Microsystems Inc. ) 技術に基づく最近の開発は上記の難点を克服することを求めたものである。Java技術は主として、(i) C及びC++に若干類似した新しいプログラミング言語、及び(ii) 仮想機械を含んでいる。本質的に、Javaプログラミング言語で作成されたプログラムはバイト・コード形式にコンパイルされてから、クライアントで実行されているJava仮想機械で実行時に解釈される。Java仮想機械はバイト・コー

ドを基礎となる物理機械によって実行できる命令に変換する。

【0010】Javaを使用して作成されたプログラムはWWWによってバイト・コードの形式でダウンロードして、クライアントにあるJava仮想機械で実行できる。このようなプログラムは「アプレット」と呼ばれている。WWWによりコードをダウンロードするためにJava技術を使用することには、2つの主要な利点がある。まず、各クライアントがJava仮想機械のコピーを持っていると想定すると、アプレットはプラットフォーム・インデペンデントである(クライアントのシステムにある仮想機械は通常、オペレーティング・システムまたはWebブラウザそのもののいずれかに組み込まれている)。換言すると、クライアントそれぞれのオペレーティング・システム及びマシンにしたがってクライアントへダウンロードするために各種のコードをサーバが持っている必要がない。したがって、単一のバージョンの関連コードだけを作成し、維持することが必要であり、これはソフトウェア開発者にとっての人生を大幅に単純化する。第2に、アプレットが物理機械ではなく、仮想機械で実行されるため、セキュリティが大幅に改善される。この場合、ネットワークによってコードをダウンロードする場合、クライアントに記憶されているデータまたはプログラムに損傷を与える何らかの不当なコード(偶然その他の)を含んでいる危険が常に存在している。仮想機械は、しかしながら、アプレットの動作を監視し、このような不当な活動を監視し、防止することができる。

【0011】ソフトウェアをバイト・コードの形式でサーバからクライアントへダウンロードして、仮想機械で実行するという概念もJava技術とは無関係であることが知られていることに留意されたい(たとえば、米国特許第5347632号参照)。

【0012】Javaアプレットを呼び出すために、HTMLテキストのWebページは、そのアプレットを含んでいるURLを特定する<APPLET>タグを含んでいる。ブラウザはアプレットを検索し、実行することによってこのタグに応答する。<PARAM>というタグも定義されており、このタグは一对の対応する<APPLET>及び</APPLET>タグの内部に含まれており、実行時にアプレットに渡されるパラメータを指定するのに使用できる。(APPLET及びPARAMタグがHTMLの標準に正式に組み込まれているものではないが、それにもかかわらず多くのWebブラウザによって認識されていることに留意されたい。) Java技術及びアプレットの詳細情報はLaura Lema及びCharles Parkinsの「Teach Yourself Java in 21 Days」(Sams. net Publishing、米国インディアナポリス、1996年)に記載されている。

【0013】このようなアプレットの大きい制限は、標準Javaモデルがアプレットに認めているのが、ダウンロード元のサーバと通信することだけであるということである。これはJava「サンドボックス」セキュリティ制限と呼ばれている。これはある種のセキュリティの利便をもたらすが、ある種のアプリケーションでJavaを使用することを大幅に制限する。たとえば、ネットワーク内の多くの他のシステムとの通信を達成することが目的であるコネクティビティが主な目的であるアプレット（ネットワークング・アプレット）には、これは望ましくない。Java開発キット（JDK）バージョン1.1などの最近のJavaのリリースはこの呼出しトラステッド・アプレットに対する解決策を提供するものであるが、この解決策はすべてのシナリオに当てはまるものではない。まず、1.02などの以前のJDKバージョンのユーザを対象としていない。第2に、最新のウェブ・ブラウザはまだJDK1.1に完全に準拠していない。第3に、最も重要なことは、ネットワーク・アドミニストレータはそのユーザがそのネットワーク中の任意のホストに接続することを望んでいないことである。その代わり、ネットワーク・アドミニストレータはアドミニストレーション管理及びセキュリティ機能という利点を備えたマルチホスト・アプレット通信の融通性を望んでいる。これらの利点のすべてを備えている解決策は入手できない。

【0014】

【発明が解決しようとする課題】本発明は発信元のサーバにおけるアプレット通信をリダイレクトすることによってアプレットが複数のホストと通信することを可能とし、発信元サーバがフィルタ及びアドオン・アドミニストレーション機能によりこのようなリダイレクトを管理する機能を備えている方法及び装置を提供する。この機能は対象アプレットも収納しているウェブ・サーバに常駐しているサーバ・アプリケーション（リダイレクタともいう）によって可能となる。「Java」及び一般的にJavaに関連した用語「アプレット」を本発明の説明に使用するが、このような用法は本発明を特定のプログラミングまたはネットワーク環境に限定することを目的とするものではない。

【0015】

【課題を解決するための手段】本発明はウェブ・サーバにインストールされ、通信を同一のサーバに常駐しているネットワークング・アプレットにリダイレクトするアプリケーション（リダイレクタ）を定義する。ネットワークング・アプレットをウェブ・サーバからダウンロードして、異なるホストに対するセッションを確立する場合、アプレット・コードはウェブ・サーバ名を検出し、リダイレクタのポート番号とのソケット接続を開く。リダイレクタは接続を確認し、ネットワークング・アプレットは遠隔ホスト・サーバ名とソケットを、これが接続

するリダイレクタに応答する。リダイレクタが要求されたホスト接続を行うと、2つの接続がネットワークング・アプレットとホストの間の通信パイプを形成し、リダイレクタは中間転送トラフィック内におかれる。このような手法により、ネットワークング・アプレットの機能を可能とするために、既存のネットワーク環境のいかなる個所の変更も必要なくなる。このようにして、ネットワークング・アプレットは遠隔ホスト及びウェブ・サーバ・アドミニストレータと通信を行うことができ、かつリダイレクタの構成を変更することにより、このようなアプレットのリダイレクトを制御することができる。

【0016】本発明はこのようなアプレットのリダイレクトを管理する手法を特に画定する。リダイレクタはホスト・アクセス・フィルタリング・テーブルによって構成される。このテーブルは管理されているすべてのホスト・アドレス、及びこれらにアクセスすることが認められているユーザ・アドレス範囲を含んでいる。たとえば、ネットワークング・アプレットがダウンロードされ、ホストAへの接続を要求している場合、アプレットが実行される機械アドレスをチェックして、これがホストAへのアクセスに認められている範囲内であるかどうかを調べる。範囲内でない場合、接続要求は拒絶され、フィルタの拒絶についての記録保存のため恐らくはログされる。「発明の実施の形態」の項で検討する例はこの対話を詳細に検討するものである。

【0017】本発明はネットワークング・アプレットにかかる負荷を検索して、いくつかのホスト・アドレスを知り、要求する静的ホスト接続のための方法も画定する。その代わりに、上述のフィルタリング・テーブルを使用して、ユーザ・アドレス範囲によりホスト接続を経路指定することもできる。リダイレクタがこの機能に合わせて構成されている場合、ネットワークング・アプレットはホスト接続が割り当てられるリダイレクタに接触することが必要なだけである。ホスト固有の接続要求なしにネットワークング・アプレットが接触した場合、リダイレクタはフィルタ・テーブルを調べて、どのホストをアプレットのアドレス範囲へデフォルトで接続するべきかを判定する。

【0018】本発明はアドオン・モジュールを使用してリダイレクタ機能を強化する方法も画定する。この手法により、負荷機能を希望する場合に、リダイレクタの機能をウェブ・サーバに再プログラムまたは再分配する必要がなくなる。その代わりに、リダイレクタがアドオン・モジュールと対話し、リダイレクタが認識する必要さえない機能を達成する手法を画定する。発明の実施の形態の項で例として検討するこのようなアドオンの1つは、クライアントからサーバへのセッション・データを暗号化するセキュリティ・アドオンのものである。このような機能により、ネットワークング・アプレットはこのような暗号化を達成するようにリダイレクタの機能を

変更する必要なしに、暗号化セッションの利益を得られる。

【0019】

【発明の実施の形態】本発明の好ましい実施の形態はJavaプログラミング・クラス及びWebクライアント／サーバ・ネットワーク環境を使用して実現される。これはユーザ作成クライアント・ネットワーキング・アプレット、ネットワーク上のターゲット・リソース及び本発明によって画定されるWebサーバに常駐するリダイレクタ・アプリケーションを含んでいる。この枠組における対話は特定の例を使用することによって最もよく理解されよう。

【0020】図1に示したネットワークの例は、現在のJavaサンドボックス・セキュリティ制限によるネットワーキング・アプレット機能を表している。Javaイネーブル・ウェブ・ブラウザまたはJavaアプレット・ビューワなどのWebクライアント(101)を使用して、ユーザはWebサーバ(103)から動的にダウンロードしたアプレット(102)を実行する。しかしながら、サンドボックス制限により、アプレットはその発信元サーバ(103)と通信を行えるだけである。サーバ(104)上のアプリケーション及びリソースをアプレットが使用して、情報を記憶したり、そのタスクを完了するのを補助したりするが、アプレットがネットワーク上にあるほかのサーバにアクセスするのを認められていない。上述したように、Java標準の最近の進歩は他のリソースにアクセスできるトラステッド・アプレットに関する枠組を定義しているが、この手法は今日のブラウザによって広くサポートされておらず、さらに重要なことは、どのリソースにアプレットがアクセスできるかについてのアドミニストレーション管理を備えていない。

【0021】図2に示すネットワークは本発明で可能なアクセスの範囲、ならびにそのアクセスに対して与えられるアドミニストレーション管理を表している。Webクライアント(201)でJavaイネーブル・ブラウザまたはJavaアプレット・ビューワを使用して、ユーザはWebサーバ(203)から動的にダウンロードしたアプレット(202)を実行する。Javaで利用可能な機能により、アプレット(202)はリダイレクタ・プロキシ(204)を含む発信元サーバ(203)上のリソースに直接アクセスできるだけである。しかしながら、リダイレクタ(204)とのこの対話から、アプレット(202)はネットワーク(205)のどこかにあるアプリケーション及びリソースに間接的にもアクセスできる。リダイレクタ(204)はアプレット(202)と接触した後、アプレットに代わって他のネットワーク・リソース(205)に接触し、アプレット(202)とこれらのリソース(205)の間で情報を送る。このようにして、アプレット(202)が利用でき

る通信の範囲は本発明のリダイレクタ(204)を使用して拡張される。この拡大したアクセスはチェックなしに与えられるものではないが、これはリダイレクタ(204)が中心に存在することによって、リダイレクタを他のネットワーク・リソースへのアクセスを管理するために使用できるからである。外部アクセスがどのように与えられるか、また管理されるかについての詳細な説明を、図3及び図4について行う。

【0022】図3は本発明のリダイレクタ(303)機能を使用して、ホスト・サーバ(305)にアクセスしているWebクライアント(301及び306)のユーザA及びBを示す。明確とするため、図の流れはJavaアプレットがWebサーバ(302)からWebクライアント(301及び306)にダウンロードされた後で始まる。通信はアプレットがWebサーバ(302)におけるそのポート番号(311)を介してリダイレクタ(303)を開くことを要求することから始まるが、図はユーザ(301及び306)がWebサーバ(302)から離隔していることを示している。これらは同一のコンピュータに常駐していてもかまわない。リダイレクタ(303)はクライアント(301)からのオープン要求を認識し、これを受け入れる(312)。クライアント(301)は次いで、Webサーバ(302)とは異なるホスト・サーバ(305)に接続することを要求する。リダイレクタ(303)はユーザのテーブル及び特権をチェックして、ユーザAがホスト・サーバ(305)(314)に接触することを認められているかどうかを調べる。このシナリオにおいては、ユーザAが接続することを認められており、リダイレクタがポートを開いて、ホスト・サーバ(305)(315)と通信するものと想定する。ホスト・サーバ(305)はこの要求(316)に正常に応答し、リダイレクタ(303)はクライアントに、ホスト・サーバ(305)(317)と通信できるようになったことを通知する。Javaサンドボックス制限がWebサーバ(302)への直接アクセスを認めるだけであり、ホスト・サーバ(305)へのこのさらなる接続は本発明によってのみ可能となることに留意されたい。この時点から、クライアント(301)は任意にホスト・サーバ(305)と情報を交換することができ、リダイレクタ(318)は仲介者(318)として活動する。

【0023】図3は適切な特権を有していないものと想定しているウェブ・クライアント(306)のユーザBも示している。クライアント(306)はリダイレクタ・ポート(319)を開き、リダイレクタ(303)は成功と応答する(320)。クライアント(306)は次いで、接続することを許可されていない遠隔ホスト・サーバ(305)に接続することを要求する(321)。リダイレクタ(303)はそのホスト・アクセス・テーブルをチェックし、ユーザBがこの接続に対して

必要な特権を有していないことを調べる(314)。リダイレクタ(303)は次いで接続要求を拒絶し、クライアント(306)にアクセスが受け入れられなかったと通知する(322)。ユーザAを受け入れ、ユーザBを受け入れないホスト・アクセス・テーブルをサーバで、あるいはホストごとにホストで遠隔で変更するか、あるいは既存のホストの流れを動的にリダイレクトするように変更することができる。このフィルタリング機能により、アドミニストレータはウェブ・クライアントに拡張アクセスを与えることができるとともに、セキュリティのための制御も維持できる。

【0024】図4は本発明によって画定されたアドオン機能を組み込んだ流れの例を示す。好ましい実施の形態で使用される特定のアドオンはクライアントとサーバの間の暗号化／暗号解読機能を可能とするものである。明確とするため、図の流れはJavaアプレットがWebサーバ(402)からWebクライアント(401)にダウンロードされた後で始まる。リダイレクタ(403)を開くことを要求しているアプレットとの通信が、Webサーバ(402)(411)におけるそのポート番号を介したセキュア接続によって始まる。安全な接続が要求されたことがわかると、リダイレクタ(403)はクライアントをそのセキュア・ポート機能(404)と接続し、成功を返す(412)。クライアント(401)は次いで、ホスト・サーバ(407)(413)に接続を要求し、リダイレクタ(403)はホスト・アクセス・テーブル(404)のユーザ特権をチェックする。ユーザがこのホスト・サーバ(407)へのアクセスが認められていることがわかると、リダイレクタ(403)はホスト・サーバ・ポート(415)を開き、正常な確認を受信し(416)、クライアントに接続の成功を通知する(417)。これでセキュア接続が確立され、クライアント(401)は暗号化されたデータ・フローをウェブ・サーバ(402)(418)に送る。セキュア・ポート(404)機能はデータ・フローが暗号化されることを知って、データを暗号化／暗号解読アドオン(406)(419)へ渡す。暗号化アドオンはシステムがサポートしている任意の形態の暗号化でよい。アドオンはデータを暗号解読し、正規のデータ・フロー(420)としてリダイレクタ(403)へ戻す。リダイレクタはデータ・フロー(421)を取り入れ、これをホスト・サーバ(407)へ送る。ホスト(407)が正常なデータをリダイレクタへ戻し、リダイレクタがこれを通して暗号化する場合にはいつでも、これと同じプロセスが逆にされる。このようにして、アプレットはサーバ・コードまたはリダイレクタ・コードへの変更な

しにセキュア・セッションを得る機能を獲得する。この例はセキュリティ・アドオンを示しているが、アドオンの概念は特定の形態のアドオンに限定されるものではなく、リダイレクタに関する管理の知識を必要とすることのない任意のその他の機能のアドオンに適用できるものである。

【0025】まとめとして、本発明の構成に関して以下の事項を開示する。

【0026】(1) 各々が一意のアドレスによって識別されるユーザ・ワークステーション及びホストのネットワークにおけるプログラムされたオブジェクトのリダイレクトを管理する方法において、アクセス対象のホストのアドレス及び前記ホストの各々にアクセスすることを認められているユーザのアドレス範囲を含んでいるホスト・アクセス・フィルタリング・テーブルを作成するステップと、前記ユーザが前記フィルタリング・テーブルにプログラム式に前記ユーザからのアクセスが認められているホストへアクセスするステップとを備えている方法。

(2) アプレットをダウンロードしたホスト以外のホストとの前記アプレットの通信を可能とする方法において、アプレットを一意のアドレスを有しているウェブ・サーバからユーザ・ワークステーションにダウンロードするステップと、前記ウェブ・サーバのアドレスを検出するステップと、前記ウェブ・サーバからリダイレクタ・ポート番号へのソケット接続を開くステップと、前記リダイレクタから前記ユーザ・ワークステーションへターゲット・ホストに対するアドレスを送るステップと、前記ターゲット・ホストと前記アプレットを実行する前記ユーザ・ワークステーションとの間のソケット接続を開くステップとを備えている方法。

【図面の簡単な説明】

【図1】サンドボックス制限(従来技術)のあるネットワーク環境の図である。

【図2】非発信元ホストへのアクセスを可能とする本発明によるネットワーク環境の図である。

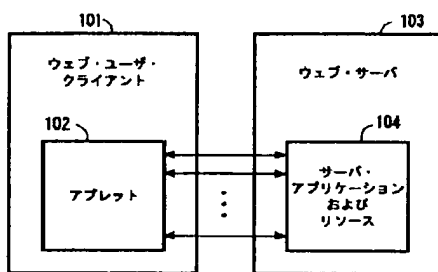
【図3】被管理拡張アクセスに対して本発明を使用した場合のアプレット、リダイレクタ及びホスト・サーバの間の流れを示す図である。

【図4】本発明による暗号化などのアドオン機能を使用するのに関与する流れの図である。

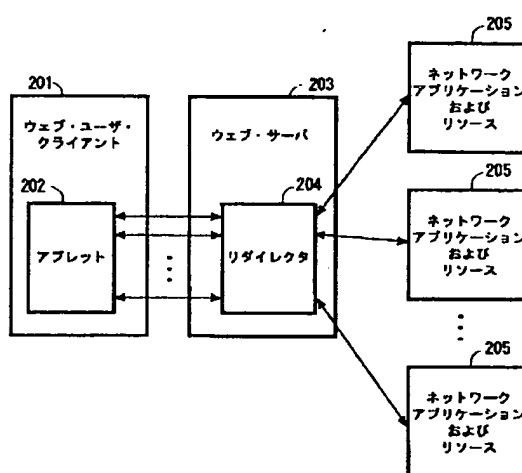
【符号の説明】

- 101 ウェブ・ユーザ・クライアント
- 102 アプレット
- 103 ウェブ・サーバ
- 104 サーバ・アプリケーション及びリソース

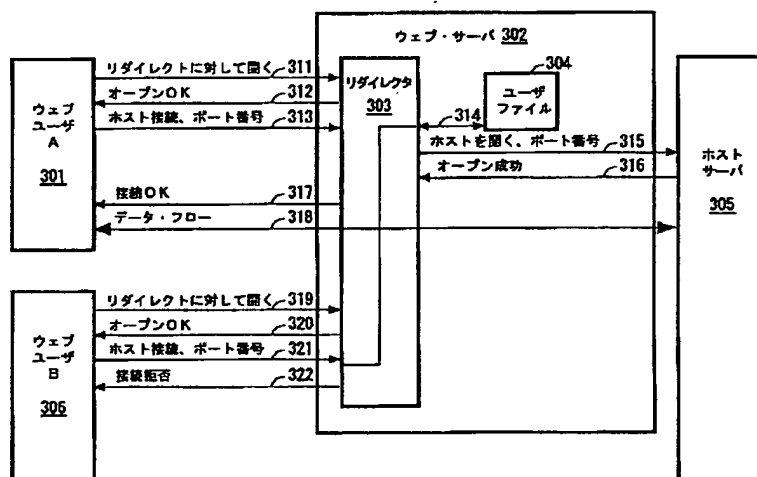
【図1】



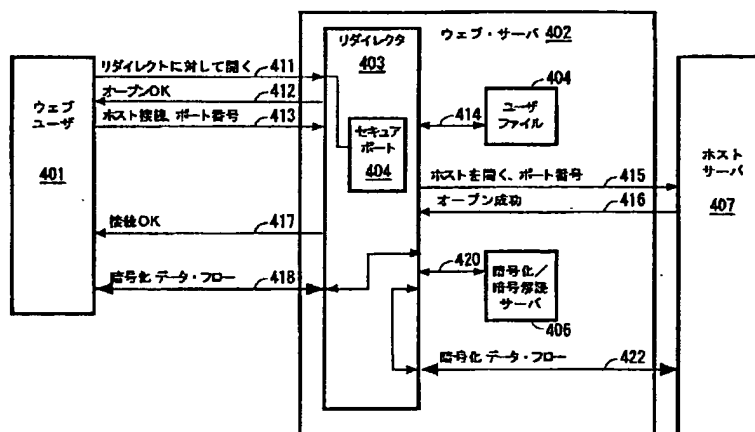
【図2】



【図3】



【図4】



## フロントページの続き

(72)発明者 デビッド・ブルース・リンクイスト  
アメリカ合衆国27613 ノースカロライナ  
州ローリー レーク・スプリングス・コート  
ト 4001  
(72)発明者 バラディック・バハライラル・ナノバティ  
アメリカ合衆国27513 ノースカロライナ  
州カリー トラファルガー・レーン 111

(72)発明者 イーシン・タン  
アメリカ合衆国27615 ノースカロライナ  
州ローリー グレットン・ブレース 300  
(72)発明者 アジャム・アキンウンミ・ウェスレイ  
アメリカ合衆国27604 ノースカロライナ  
州ローリー カーディナル・ドライブ  
500